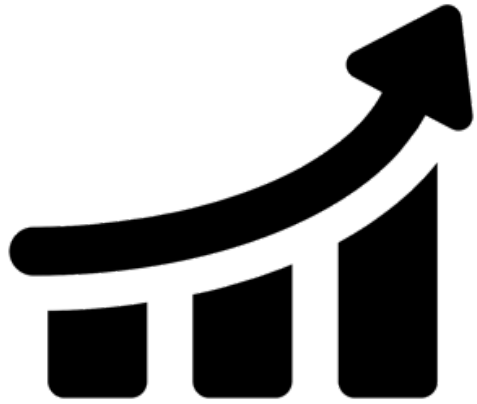


Moving the web ahead with open tooling





The web is a
roaring success...



- It is hard to imagine a world without the world wide web
- Web technologies are standardised and browser differences are negligible
- Open source has matured from a niche to the technology that runs the web
- Documentation is plentiful, free and interactive



So, why is it all
kind of a mess?



- Blatant lack of accessibility
- Security problems
- Performance issues
- A need for blockers to keep the web safe and a lot less annoying



No shortage of
opinions why that is



- Users are terrible and have horrible setups
- The web stack is just not good enough
- Greedy corporations plaster the web with ads and see users as sheep to milk (?)
- We don't have the right tools
- Tech bros build horrible things without consideration
- Aliens



Sticking to the
ratio...



We have 2 ears and 1
mouth – maybe we
should spend more time
listening and less talking.



I'm in the middle
of a launch...

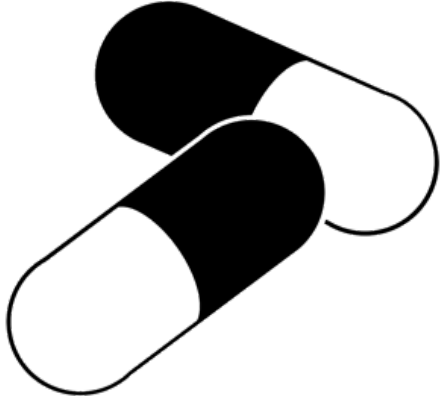


User needs > opinions

aka.ms/edgeinsider

Different users, different needs

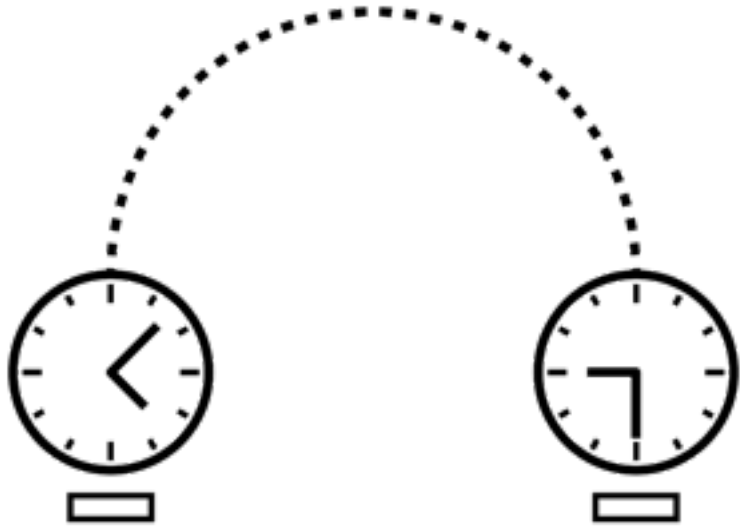




Somehow bitter
pills to swallow



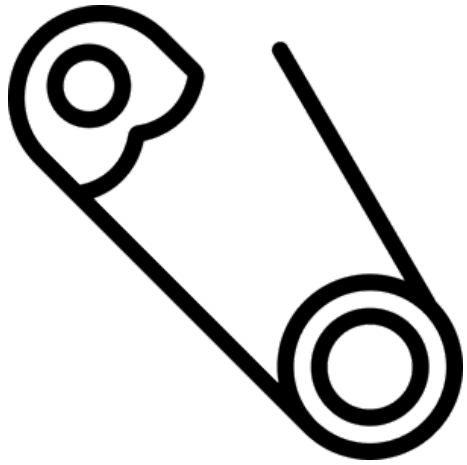
- To different users, a browser is
 - something they can't control
 - something they cherish and highly customize
 - irrelevant, as all of them work OK enough
 - nothing worth changing their habits for
- Security and privacy aren't things that people control, but trust in the platform to do for them



Times changed...



- The web isn't a cool new thing
- It's akin to plumbing
- The shiny fixtures are mobile devices and "smart" products



A fix seems to be
somewhere else...



- It might be up to us to fix the web
- This means taking ownership and caring for the needs of different constituencies
- We can't expect users to care enough to fix the web – we need to empower developers, maybe?

Different developers, different needs



“This is a job” developer



I work for a company that defines our tools and computer setup. I know there are probably better things out there, but as I don't get to use them at work, why bother learning them?

The “switched on” developer



It is frustrating that there are still people out there with old computers and outdated browsers. With a library I can support them as I'd rather learn and try out new things.

The “I am here to deliver” developer



My clients don't pay extra for fancy new technology. They need to get something that works on their machines. It needs to be legally compliant, you know, accessible so they can't be sued. That's why I use a framework.

The “the web is broken” developer

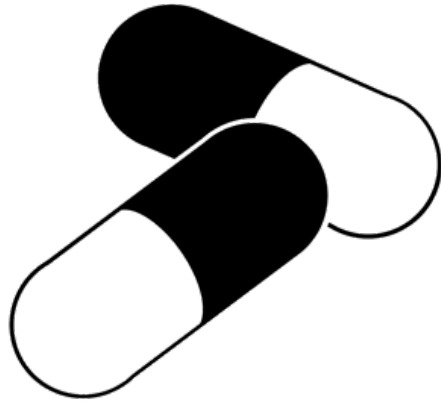


I’ve always worked with higher programming languages and rich UI environments. The web seems like a mess and I don’t want to spend time trying to guess what the outcome of my code is.

The “I’m here to learn” developer



I just started as a web developer and wherever I look, I am confused. There is so much information and often people give contradicting advice. I just want to learn the right thing...



Somehow bitter pills to swallow (2)



- The caring developer is a tiny fraction of the market
- For many, this is a job to make a living
- Satisfying developer needs has no direct correlation to creating better user products
- New developers don't find an empathic, open world, but an avalanche of choice and opinions



Why learn the basics?



- Browsers can't break the web – their forgiveness knows no bounds
- If it works, why bother learning why?
- If it doesn't work, there's ample of resources to learn quick fixes.



Tech innovation
isn't the answer



- Constant change in the developer space is a double-edged sword
- We empower a small group of highly motivated people at the cost of overwhelming others
- We innovate to fill a specialist need and call it a success because it is new instead of solving real problems
- We don't find out about real problems as we don't ask people outside our bubble



We <3 tools!



- Browser developer tools
- Editors, extensions and themes
- Terminals and their configuration
- Deployment and optimization toolchains
- Abstractions, Components and libraries



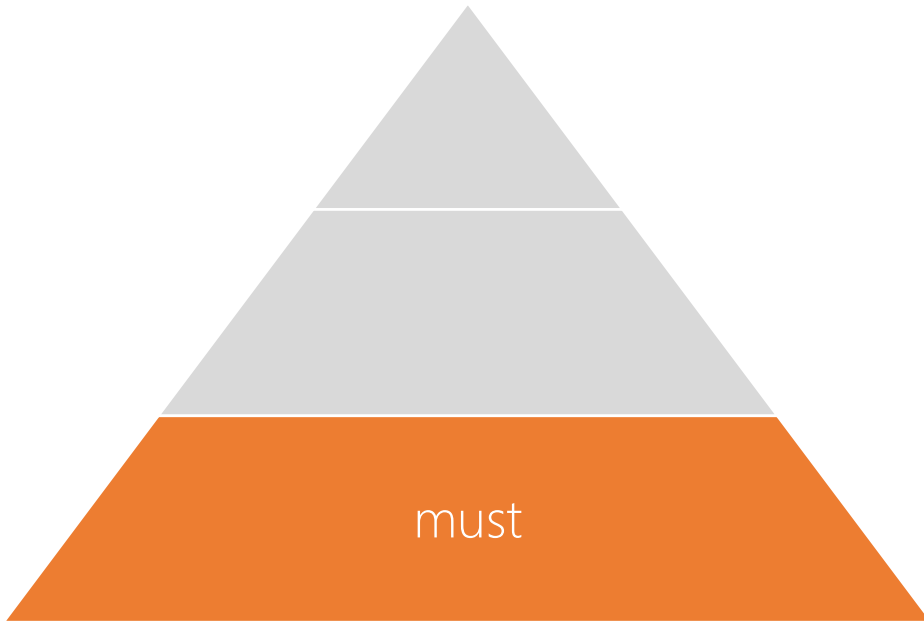
It is all a bit
much...



- We overwhelm people who want to start with the web – or just want to do their job
- By demanding people to set up complex dev environments we inhibit diverse contribution
- We don't focus enough on human needs in our tools

Focusing on delivery

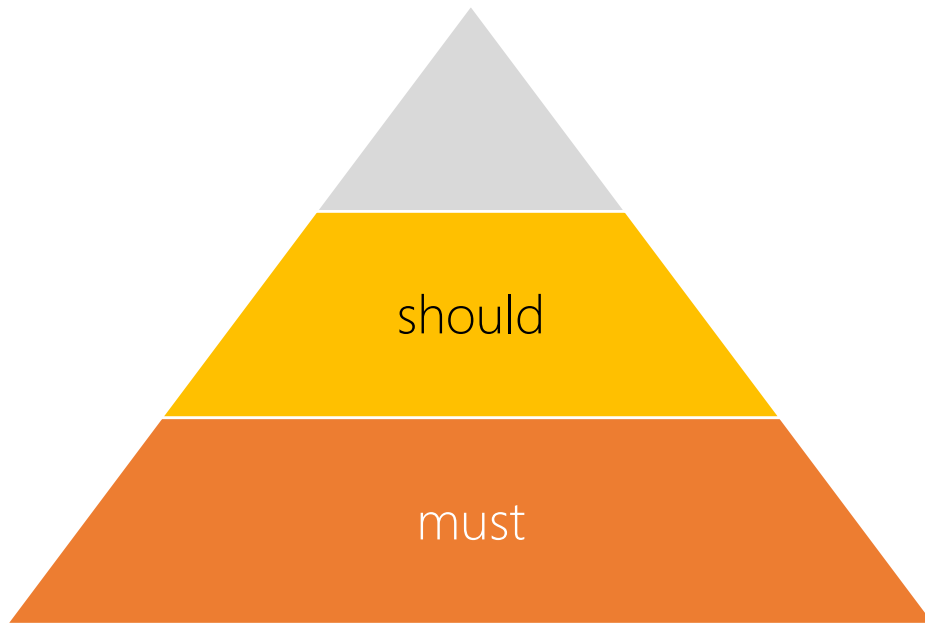




Filling the needs



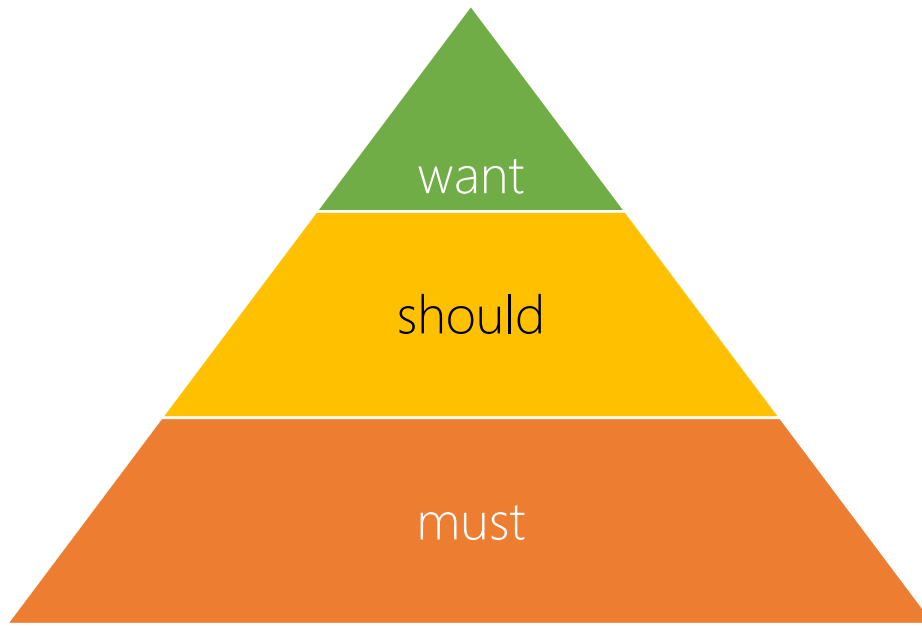
- Availability
- Accessibility
- Security
- Mutability to user needs



Adding quality



- Enhanced performance
- Extended availability (offline)
- Customisability



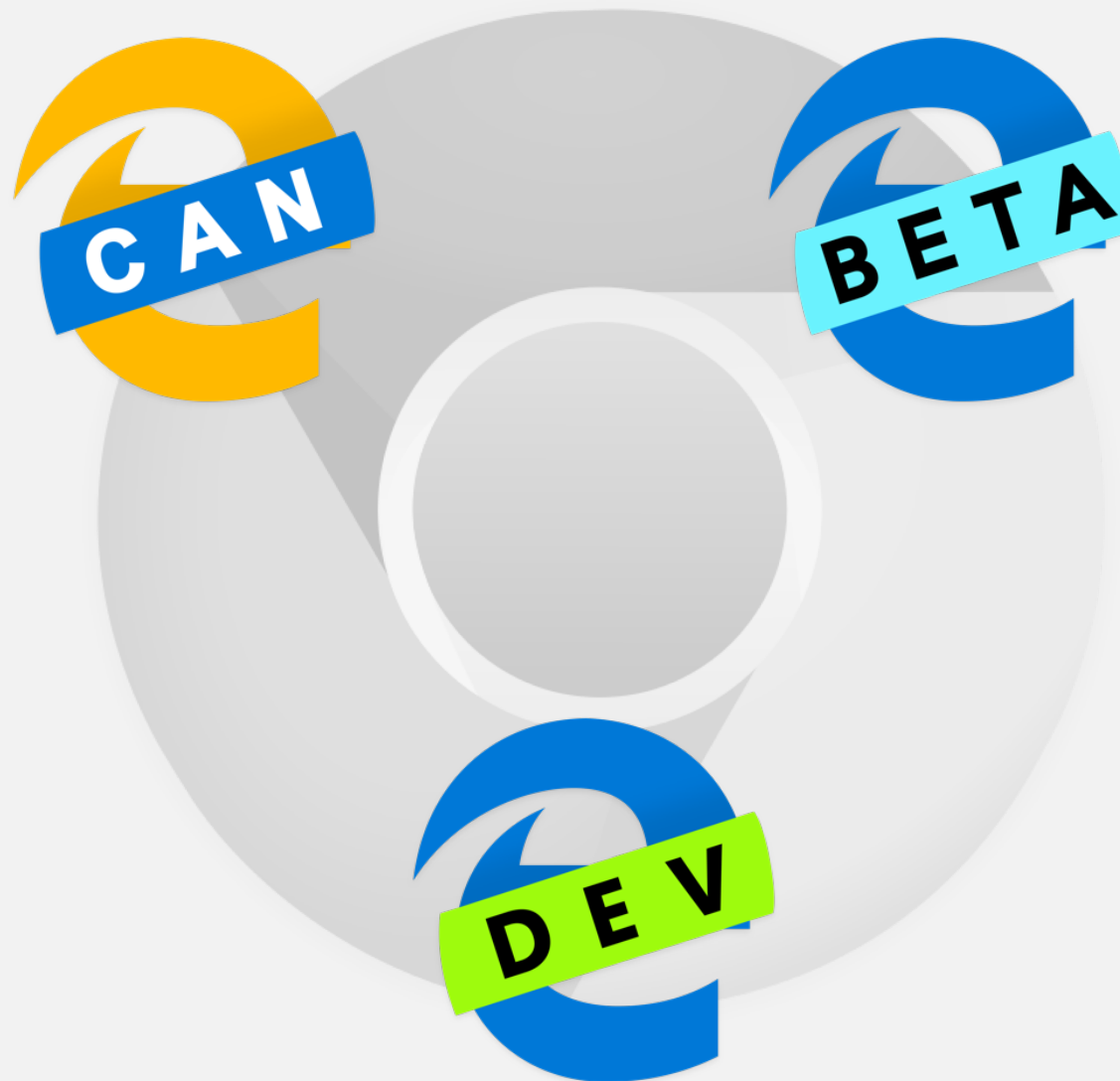
Creating delight



- Make full use of the platform
- Predict use cases and prevent simple mistakes
- Offer extended functionality



Edge on Chromium



aka.ms/edgeinsider



Edge on Chromium

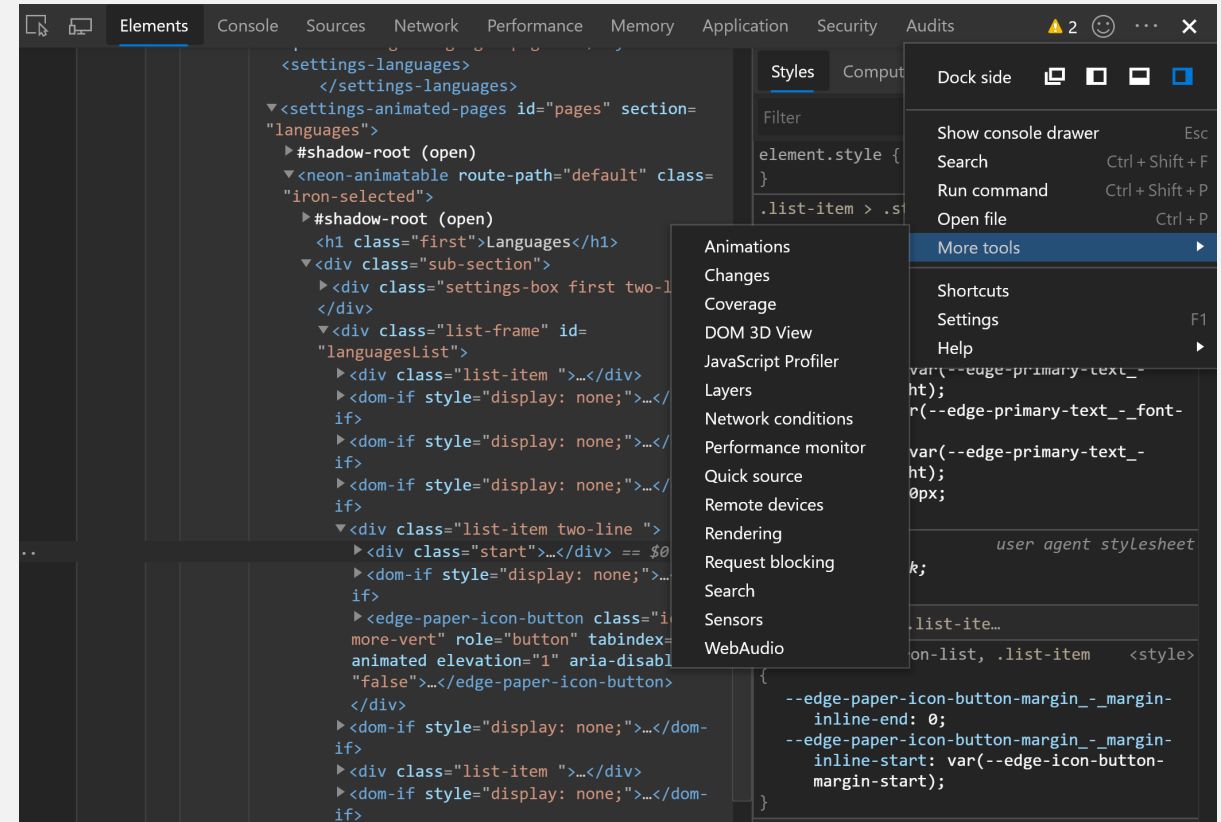


- Adding a powerful player with lots of users to an important part of the web
- Lots of bug fixes
- Accessibility enhancements to browser and devtools
- Internationalisation of browser and developer tools (30% of VSCode users change the language)

aka.ms/edgeinsider



Hypothesis: tools
are too complex



Building upon and connecting great tools...

Switching to Chromium is a pivot point for Edge DevTools



The tooling ecosystem has never been better

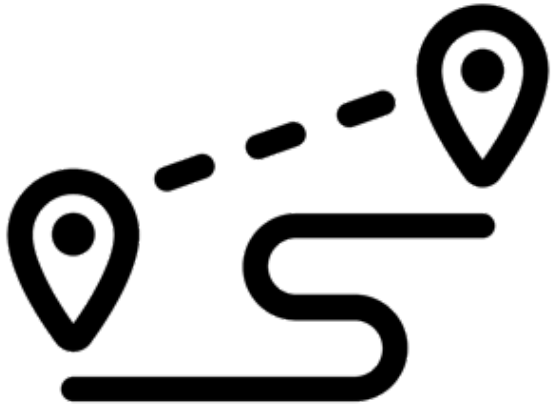


But we know from feedback it can get even better



Opportunity to build on OSS, integrate naturally, and keep learning





Tools to help
simplify webdev





Node: the glue



- JavaScript on the server
- Enables a highly customisable web infrastructure (servers, build tools...)
- Elevated JavaScript from “how does browser \$x break” to a solution for all kind of web tasks



Chromium: the engine



- Open Source
- Multi Platform
- Powers various browsers and App engines (Electron)
- Can be automated and scripted (Puppeteer)



Visual Studio Code: the “hackable” editor



- Open Source
- Multi Platform
- Extensible / Customisable
- Based on Electron/Chromium
- Includes Git version control
- Has built-in terminal
- Lightweight, code focused

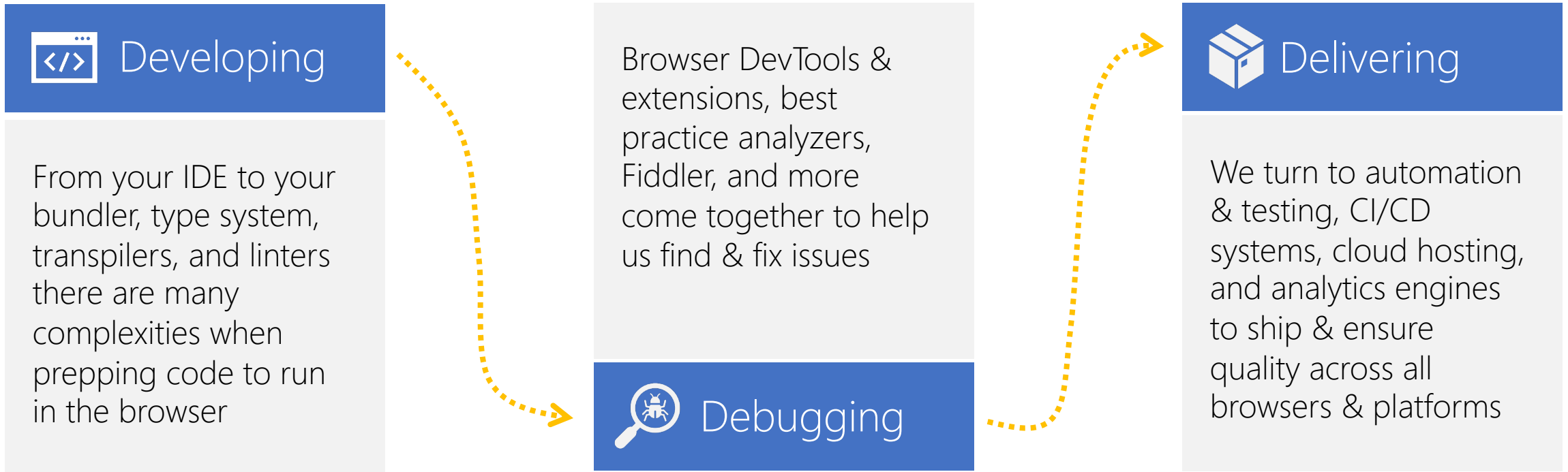


Webhint: testing and best practices





- Testing tool to check web projects for:
 - Accessibility issues
 - Performance problems
 - App readiness
 - Compatibility with standards
 - Common pitfalls to avoid
 - Security issues
- Online service (webhint.io) or node package to integrate in other tools
- Highly customisable

From development to delivery





From development to delivery



Develop

 Code
+
 webhint

aka.ms/webhint4code



 Code
+
 Elements

aka.ms/elements4code



 Code
+
 Debugger

aka.ms/debugger4code



Debug

 + 
DevTools



aka.ms/edgeinsider

 webhint
+
 DevTools

aka.ms/webhint4edge


 webhint
+
 DevTools

aka.ms/webhint4chrome



 webhint
+
 DevTools

aka.ms/webhint4firefox


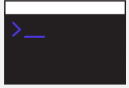
Deliver

 WebDriver


aka.ms/msedgedriver

 +  Puppeteer

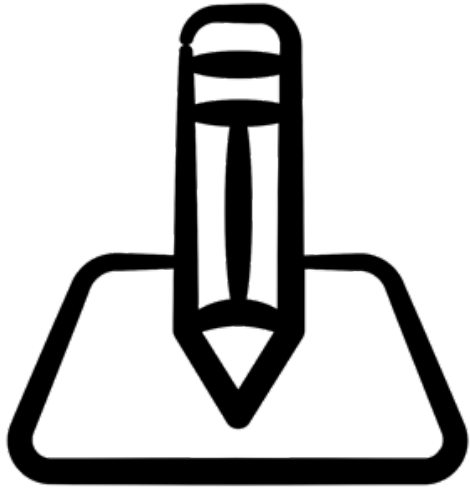
Coming soon!

 webhint
+


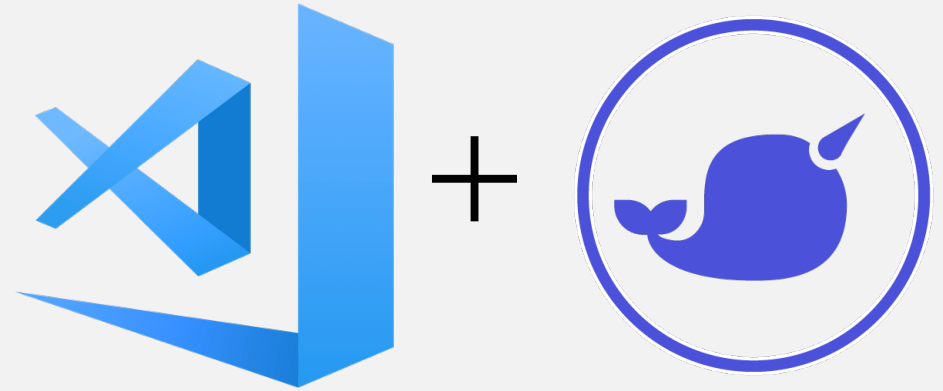
aka.ms/webhint4node

 webhint
Livesite Scanner

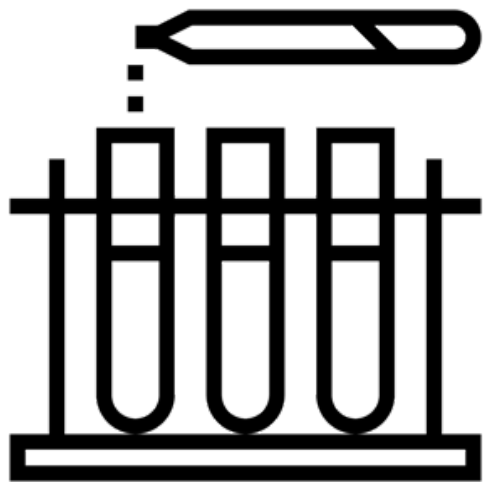
webhint.io



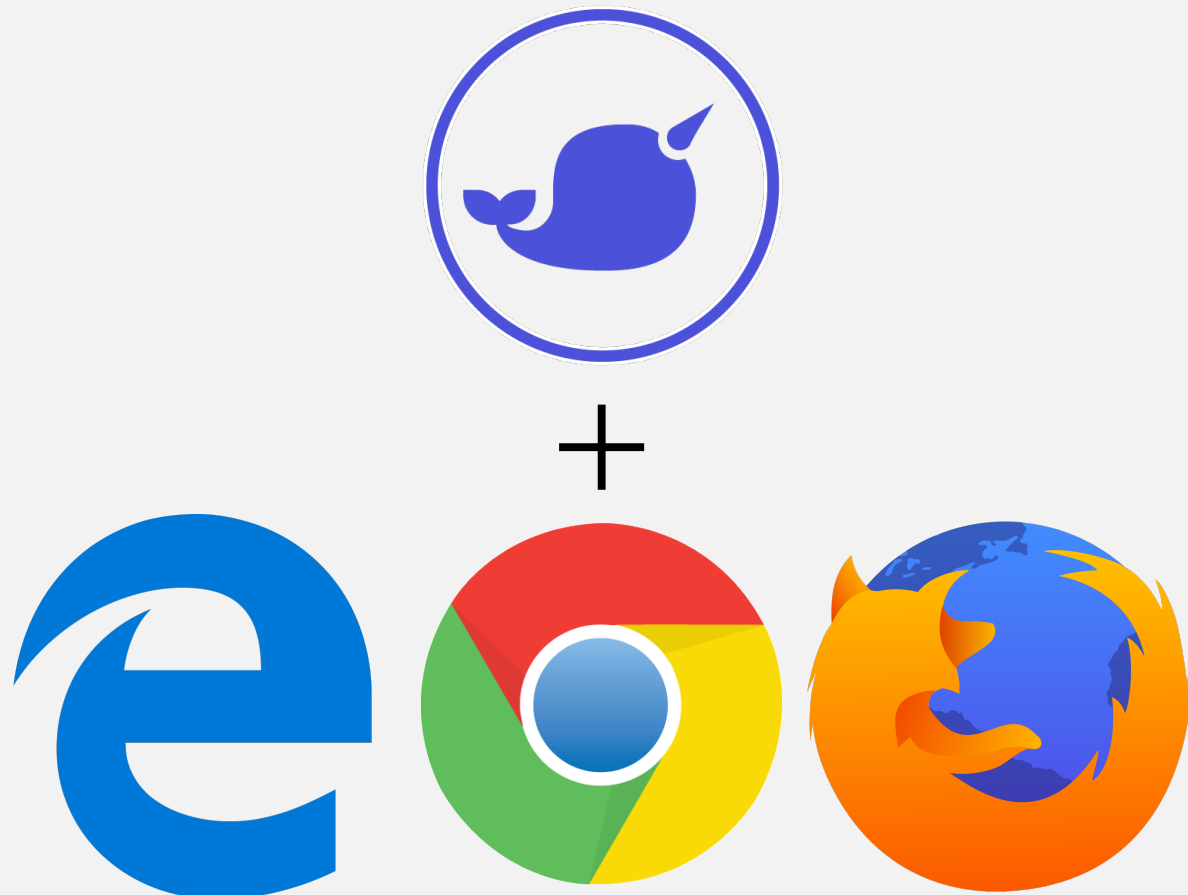
Show problems with
your code while you
write it...



aka.ms/webhint4code



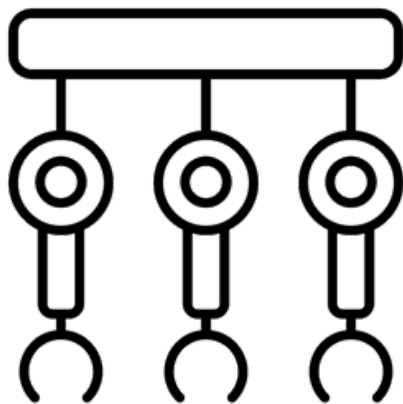
Test your sites in
the browser...



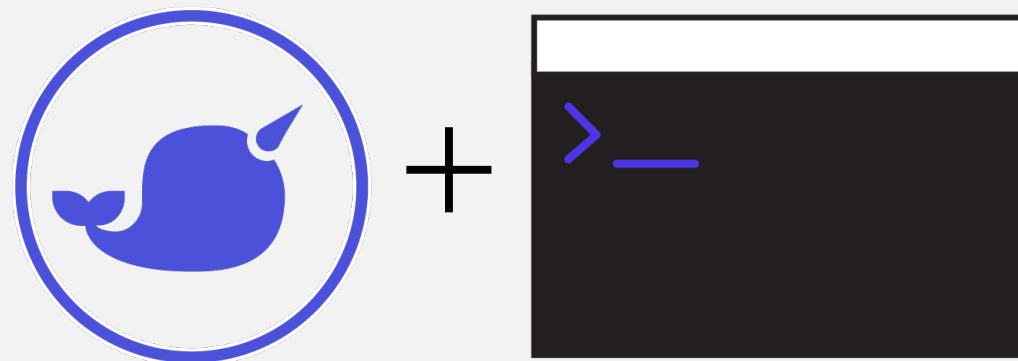
aka.ms/webhint4edge

aka.ms/webhint4firefox

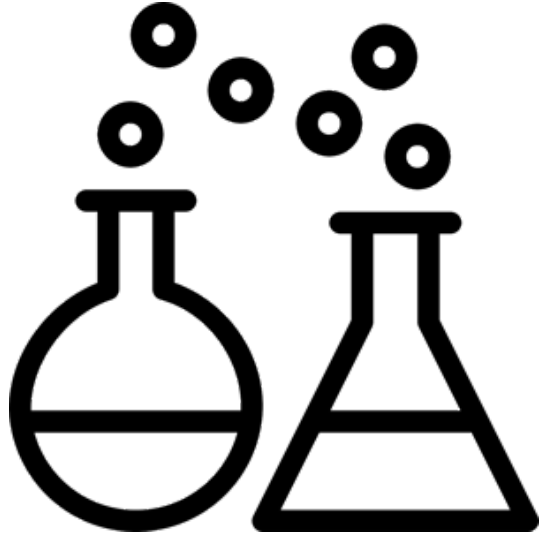
aka.ms/webhint4chrome



Add automated
testing to your build
process...



aka.ms/webhint4node

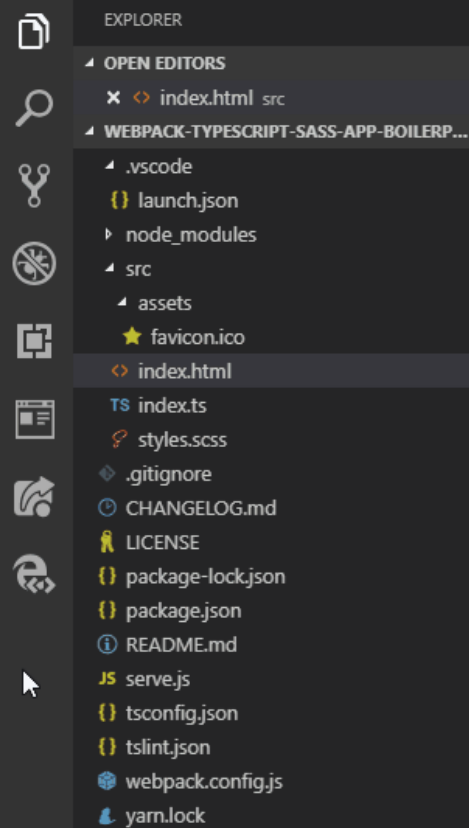


Elements for Code



- Edge/Chromium developer tools inside VS Code
- No need to switch to the browser to try a few things out
- No need to replicate the same functionality in both places

aka.ms/elements4code



<> index.html x

src > <> index.html > html > body

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Web App</title>
6 </head>
7 <body>
8   <div class="container">
9     <div class="banner container">
10      <h1>Elements for Microsoft Edge</h1>
11    </div>
12    <button id="addButton">Add a new item</button>
13    <div class="container" id="list"></div>
14  </div>
15 </body>
16 </html>
17
```

What's next?





What's missing on
the web platform?



webwewant.fyi

In summary...



- The web is a diverse place
- We have a chance to make it easier for everyone to create quality products for all end users
- We have open source platforms with open communication channels that power the web
- We accumulated a lot of knowledge, why not add it to our tools to teach the creators of the web in context?

Thanks – your voice matters!



Chris Heilmann

@codepo8

@EdgeDevTools

christianheilmann.com

Click the 😊 in Edge developer tools!

